POLITECHNIKA GDAŃSKA

Karta przedmiotu

Nazwa i kod przedmiotu	Obiektowe języki programowania III, FIZ1C313		
Kierunek studiów	Fizyka Techniczna		
Data rozpoczęcia studiów	październik 2014 r.	Rok akademicki realizacji przedmiotu	2016/2017
Poziom kształcenia	I stopnia - inżynierskie	Grupa zajęć	
Forma studiów	stacjonarne	Sposób realizacji	na uczelni
Rok studiów	3	Język wykładowy	polski
Semestr studiów	5	Liczba punktów ECTS	6.0
Profil kształcenia	ogólnoakademicki	Forma zaliczenia	zaliczenie
Jednostka prowadząca	Wydział Fizyki Technicznej i Matematyki Stosowanej -> Katedra Fizyki Ciała Stałego		
Imię i nazwisko wykładowcy (wykładowców)	Odpowiedzialny za przedmiot	Łukasz Rybka	
	Prowadzący zajęcia z przedmiotu	prof. dr hab. Julien Guthmuller Łukasz Rybka dr hab. Jan Franz	

Formy zajęć i metody nauczania							
	Forma zajęć	Wykład	Ćwiczenia	Laboratorium	Projekt	Seminarium	RAZEM
	Liczba godzin zajęć	15.0	0.0	60.0	0.0	0.0	75
	W tym liczba godzin zajęć na odległość: 0.0						

Aktywność studenta i liczba godzin pracy					
	Aktywność studenta	Udział w zajęciach dydaktycznych, objętych planem studiów	Udział w konsultacjach	Praca własna studenta	RAZEM
	Liczba godzin pracy studenta	75	8.0	67.0	150

Cel przedmiotu	Utrwalenie motywacji programowania obiektowego. Powtórzenie podstawowych zagadnień programowania obiektowego na przykładzie C++ i Javy. Zapoznanie z podstawami języka Java. Zapoznanie z podstawowymi idiomami programowania obiektowego (RAII, wyjątki). Wstęp do zagadnień generyczności.

Efekty uczenia się przedmiotu

Efekt kierunkowy	Efekt z przedmiotu	Sposób weryfikacji i oceny efektu
[K_K02] Ma świadomość własnych ograniczeń i wie, kiedy zwrócić się do ekspertów.	Student jest w stanie określić podzbiór wiedzy o programowaniu zorientowanym obiektowo, z którym zapoznał się w trakcie wykładu.	[SK2] Ocena postępów pracy [SU2] Ocena umiejętności analizy informacji
[K_W05] Posiada podstawową wiedzę w zakresie metodyki i technik programowania oraz wykorzystywania wybranych narzędzi informatycznych w fizyce i technice.	Student jest w stanie zaprojektować nieskomplikowaną hierarchię klas z wykorzystaniem dziedziczenia publicznego. Odróżnia dziedziczenie od agregacji. Student potrafi stosować wyjątki, pracuje zgodnie z idiomem RAII. Student zna podstawy programowania generycznego, wprawnie operuje generycznymi kontenerami biblioteki standardowej.	[SK2] Ocena postępów pracy [SW1] Ocena wiedzy faktograficznej
[K_K01] Rozumie potrzebę uczenia się przez całe życie oraz potrzebę podnoszenia kompetencji zawodowych i osobistych. Potrafi inspirować i organizować proces uczenia się innych osób.	Student zdaje sobie sprawę z koniecznej wybiórczości zagadnień poruszanych na wykładzie. Potrafi wyszukać źródła pozwalające pogłębić posiadaną wiedzę.	[SU4] Ocena umiejętności korzystania z metod i narzędzi [SK5] Ocena umiejętności rozwiązywania problemów występujących w praktyce [SU2] Ocena umiejętności analizy informacji
[K_U03] Posiada umiejętność programowania w wybranym języku oraz stosowania podstawowych pakietów oprogramowania.	Student jest w stanie zaprojektować i zaimplementować nietrywialne programy w języku C++, proste programy w języku Java. Rozumie różnice między standardem ISO języka C++ a dialektami oferowanymi przez dostawców kompilatorów. Student wprawnie posługuje się pakietem Visual Studio i środowiskiem Eclipse.	[SW1] Ocena wiedzy faktograficznej [SU4] Ocena umiejętności korzystania z metod i narzędzi
[K_U01] Potrafi uczyć się samodzielnie, pozyskiwać informacje z literatury, baz danych oraz innych właściwie dobranych źródeł.	Student jest w stanie pogłębić swoją wiedzę korzystając z zasobów czytelni, biblioteki, internetu, sam potrafi dokonać selekcji materiałów źródłowych.	[SW1] Ocena wiedzy faktograficznej [SU1] Ocena realizacji zadania

Treści przedmiotu

Motywacja programowania obiektowego, klasa, obiekt, instancja, pola, metody, sekcje private, protected, public, ukrywanie, interfejs a implementacja, akcesory, modyfikatory, niezmienniki, konstruktor, lista inicjalizacyjna. Konstruktor domyślny, syntetyzowany, kopiujący. Przeciążanie. Const-correctness. Pola i metody statyczne. Dziedziczenie publiczne. Zasada podstawialności Liskov. Self i super. Zasłanianie, delegowanie. Serializacja. Wyjątki: motywacja, rozważania praktyczne. Strumienie w C++. Stan zepsuty, detekcja końca pliku. Podział projektu na pliki. Pliki nagłówkowe. Przestrzenie nazw. Zasada jednokrotnej definicji. Strażniki włączania. RAII w C++. Analogi RAII w Javie: try-catch-finally, try-with-resources. Podstawy języka Java, final, abstract, public, interfejsy, klasy kopertowe, referencje, garbage collector. Java vs C++. Programowanie generyczne. Kolekcje i kontenery. Szablony klas i funkcji, konkretyzacja, specjalizacja, trudności modelu C++, type erasure. Przegląd nowości C++11.

Wymagania wstępne i dodatkowe

Student zna podstawy działania komputerów o architekturze Von Neumanna. Student zna podstawy programowania w języku C++. Student ZALICZYŁ przedmiot „Obiektowe języki programowania” na sem. 4.

Sposoby i kryteria oceniania osiąganych efektów uczenia się

Sposób oceniania (składowe)	Próg zaliczeniowy	Składowa oceny końcowej
zaliczenie laboratorium	51.0%	50.0%
zaliczenie wykładu	63.0%	50.0%
dodatkowe punktu za frekwencję	0.0%	0.0%

Zalecana lista lektur

Podstawowa lista lektur	1. Bjarne Stroustrup, Język C++, Wyd. Nauk. Techn., 2008. 2. Stanley Lippman, Josee Lajoie, Podstawy języka C++, Wyd. Nauk. Techn., 2001. 3. Nicolai M. Josuttis, C++: biblioteka standardowa. Podręcznik programisty., Helion, 2003. 4. Bruce Eckel, Thinking in C++, Helion, 2002. 5. Brucke Eckel, Thinking in Java, Helion, 2006.
Uzupełniająca lista lektur	1. David Vandevoorde, Nicolai M. Josuttis, C++ szablony: Vademecum profesjonalisty, Helion, 2003. 2. Herb Sutter, Andrei Alexandrescu, Język C++: Standardy kodowania, Helion, 2005
Adresy eZasobów	Adresy na platformie eNauczanie:

Przykładowe zagadnienia/ przykładowe pytania/ realizowane zadania	1. Motywacja programowania obiektowego. 2. Klasa, obiekt, instancja, pola, metody, sekcje private, protected, public, ukrywanie, interfejs a implementacja, akcesory, modyfikatory, niezmienniki. 3. Konstruktor, lista inicjalizacyjna. Konstruktor domyślny, syntetyzowany, kopiujący. 4. Przeciążanie. 5. Const-correctness. 6. Pola i metody statyczne. 7. Dziedziczenie publiczne. Zasada podstawialności Liskov. Self i super. Zasłanianie, delegowanie. 8. Serializacja. 9. Wyjątki: motywacja, rozważania praktyczne. 10. Strumienie w C++. Stan zepsuty, detekcja końca pliku. 11. Podział projektu na pliki. Pliki nagłówkowe. Przestrzenie nazw. Zasada jednokrotnej definicji. Strażniki włączania. 12. RAII w C++. Analogi RAII w Javie: try-catch-finally, try-with-resources. 13. Podstawy języka Java, final, abstract, public, interfejsy, klasy kopertowe, referencje, garbage collector. Java vs C++. 14. Programowanie generyczne. Kolekcje i kontenery. Szablony klas i funkcji, konkretyzacja, specjalizacja, trudności modelu C++, type erasure. 15. Przegląd nowości C++11.
Praktyki zawodowe w ramach przedmiotu	Nie dotyczy

Dokument wygenerowany elektronicznie. Nie wymaga pieczęci ani podpisu.